SIAM/ASA J. Uncertainty Quantification
Vol. 5, pp. 694–719

Efficient History Matching of a High Dimensional Individual-Based HIV Transmission Model*

Ioannis Andrianakis†, Nicky McCreesh†, Ian Vernon‡, Trevelyan J. McKinley§, Jeremy E. Oakley¶, Rebecca N. Nsubuga‖, Michael Goldstein‡, and Richard G. White†

Abstract. History matching is a model (pre-)calibration method that has been applied to computer models from a wide range of scientific disciplines. In this work we apply history matching to an individual-based epidemiological model of HIV that has 96 input and 50 output parameters, a model of much larger scale than others that have been calibrated before using this or similar methods. Apart from demonstrating that history matching can analyze models of this complexity, a central contribution of this work is that the history match is carried out using linear regression, a statistical tool that is elementary and easier to implement than the Gaussian process–based emulators that have previously been used. Furthermore, we address a practical difficulty with history matching, namely, the sampling of tiny, nonimplausible spaces, by introducing a sampling algorithm adjusted to the specific needs of this method. The effectiveness and simplicity of the history matching method presented here shows that it is a useful tool for the calibration of computationally expensive, high dimensional, individual-based models.

Key words. emulation, calibration, Gaussian processes, linear regression

AMS subject classifications. 62-07, 62P10, 62J05

DOI. 10.1137/16M1093008

1. Introduction.

Approximately 1.5 million people died from AIDS-related illnesses in 2013, with sub-Saharan Africa accounting for 74% of those deaths [24]. In the same year, 2.1 million people were newly infected with HIV. Although both HIV incidence and mortality have fallen in recent years, more intensive treatment and control strategies are needed to accelerate the decline. Antiretroviral therapy (ART) is known to suppress the virus and stop

*Received by the editors September 7, 2016; accepted for publication (in revised form) March 27, 2017; published electronically August 1, 2017.

http://www.siam.org/journals/juq/5/M109300.html

Funding: This work was funded by the UK Medical Research Council (MRC) and by the UK Department for International Development (DFID) under the MRC/DFID Concordat agreement that is also part of the EDCTP2 program supported by the European Union (MR/J005088/1). The work of the eighth author was additionally supported by the Bill and Melinda Gates Foundation (TB Modelling and Analysis Consortium: OPP1084276) and by UNITAID (4214-LSHTM-Sept15; PO #8477-0-600).

†Department of Infectious Disease Epidemiology, London School of Hygiene & Tropical Medicine, London WC1E 7HT, UK (andrianakis@yahoo.com, nicky.mccreesh@lshtm.ac.uk, richard.white@lshtm.ac.uk).

‡Department of Mathematical Sciences, Durham University, Durham DH1 3LE, UK (ian.vernon314@gmail.com, Michael.Goldstein@durham.ac.uk).

§College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter EX4 4QF, UK (t.mckinley@exeter.ac.uk).

¶School of Mathematics and Statistics, University of Sheffield, Sheffield S3 7RH, UK (j.oakley@sheffield.ac.uk).

‖Medical Research Council/Uganda Virus Research Institute, Uganda Research Unit on AIDS, Entebbe, Uganda (rebecca.nsubuga@mrcuganda.org).

the progression of the disease, and it can also prevent onward transmission. This therapy is available in various sub-Saharan countries and is typically administered when the CD4 T-cell count of a patient falls below a threshold. There is, however, an ongoing discussion about removing this threshold and the effect such a policy would have on the general population.

Modeling offers one way of studying the effects of different interventions. An individual-based model (simulator) has been developed at the London School of Hygiene and Tropical Medicine which can simulate HIV transmission and the effects of ART and predict the effect of different interventions over a horizon of 10–15 years. The simulator has a number of input parameters, the values of which are uncertain, and this uncertainty should be included in any predictions we wish to make. The availability of historical data (observations) allows us to learn about the values of the input parameters, by *calibrating* the simulator to the observations. By calibrating we mean finding a subset of input parameter values for which the simulator's outputs closely match historical observations on demography, HIV prevalence, mortality, etc.

Calibrating this simulator is challenging, mainly due to the large number of input (96) and output (50) parameters. Having to simultaneously match a large number of outputs means that there are many constraints that need to be satisfied, and this can result in a very small region of the input space where the simulator matches the observations. In high dimensional input spaces, the search for a small part that will generate output matches can require a prohibitively large number of simulator runs. A further complication in our case is that the simulator is stochastic; therefore repeated evaluations are required for the same input values to estimate the mean values of the outputs.

A large number of methodologies for calibrating simulators are available, which include simple least-squares techniques, Markov chain Monte Carlo (MCMC)-based methodologies [7, 19], particle filters [3], and approximate Bayesian computation (ABC) [23, 17]. For various reasons these methodologies are extremely difficult to apply in our case. Data augmentation approaches would require reconstruction of the likelihood function and integration over a very large hidden state space, while simulation-based methods would require a very large number of simulator runs. The latter have only really been applied to relatively small scale simulators (in terms of the number of inputs and outputs; usually around 5–10 in each case).

The problem of calibrating a simulator could also be thought of as an optimization problem: finding simulator inputs to minimize the difference between the simulator outputs and some observed target values. The optimization of an expensive simulator with a univariate output is considered in [10], and an extension for the multivariate case is given in [12]. Again, we think these approaches would be difficult to apply in our case, given the high dimensional input and output, and it is not obvious how one would account for simulator input uncertainty if the aim were simply to find a "best" value.

History matching [6] is a (pre-)calibration method that has been applied with success to slow simulators with typically larger numbers of inputs/outputs than the simulators used in the methods mentioned above. History matching tries to identify those parts of the simulator's input space where, if evaluated, the simulator is likely to match the observations. This goal is achieved via identifying regions of the input space where a match is unlikely to be found (these regions are known as *implausible*) and discarding them in iterations known as *waves*. History matching can deal with simulators that are slow to evaluate by employing statistical

models of the simulator (known as *emulators*), whose key characteristic is a trivial evaluation time.

History matching was first applied in the field of oil simulator modeling [6] and has since found applications in areas as diverse as galaxy formation [26, 28], environmental models [9], systems biology [25, 29], ocean modeling [32], and epidemiology [2]. The dimensionality (i.e., number of inputs (P) and outputs (R)) of the simulators analyzed in those works was considerably smaller; for example, [26, 28] ($P = 17, R = 11$), [9] ($P = 17, R = 13$), [25] ($P = 8, R = 15$), [32] ($P = 26, R = 4$), and [2] ($P = 22, R = 18$). An exception is [6], which studied an oil reservoir model with $P = 40$ and $R = 77$. However, [6] used the technique of active inputs to perform a substantial dimensional reduction of the simulator, showing that it is accurately described by a series of outputs possessing only three input dimensions each. In contrast, many high dimensional simulators, such as the HIV model we are concerned with here, possess a far more intricate input-output structure, for which such a large dimensional reduction is not possible.

In this work we apply history matching to a stochastic agent–based simulator with more input and output parameters than any other simulator that has been calibrated before with this or with other methods that we know of. A key contribution of this work is that we calibrate the simulator using elementary statistical tools and methodologies, which should be known to all statisticians and most modelers with basic statistical training.

The emulators used in history matching are typically built using Gaussian processes (GPs) [2]. In our experience, we have often found this to be an obstacle in applying the method, as not everyone is familiar with this elegant but nontrivial statistical model. The emulators we use in this work are based on linear regression models, which are much easier to code, fit, and interpret. History matching typically requires repeated fitting of emulators, either multivariate or a large number of univariate ones at each wave. Because its emphasis is on excluding the implausible input space iteratively, it does not depend on the availability of very precise emulators to achieve this; the same result can sometimes be achieved with less precise emulators and a few additional iterations. Therefore, linear regression models are likely to be sufficient, especially in the first waves of a history match. While we do not argue against the use of more complex statistical models for building emulators, we demonstrate that even a simple and well-known tool, such as linear regression, can be used to make considerable progress in calibrating complex and computationally demanding simulators. Furthermore, the history matching framework facilitates the use of various statistical models at different stages of the process. Therefore, linear regression can be used in the initial waves, and more complex regression tools, such as GPs, can be introduced later on, if some outputs are hard to emulate or greater accuracy is required.

Another contribution of this work is the proposal of an algorithm that can uniformly sample the nonimplausible space. After several waves of history matching, the remaining nonimplausible space can be a tiny proportion of the original (i.e., at wave 0). In general, there is no analytical description of this space, and the only way to describe it is via sampling. However, this can be challenging, since this space has as many dimensions as the simulator's inputs, has an unknown (perhaps multimodal) shape, and can be several orders of magnitude smaller than the original input space. The algorithm we propose is based on the slice sampler, is straightforward to implement, and takes advantage of the specific needs of history matching

to increase its efficiency. Another sampling algorithm that addresses the same problem has been proposed in [33], although that algorithm is intricate and significantly more challenging to implement.

History matching can give valuable insight into a simulator's structure and the structure of the constraints imposed by the data. For example, by studying the correlation patterns of the fitted inputs and outputs, one can understand or verify how the simulator's internal processes interact, information that could be useful in developing the simulator further or in deriving model discrepancy terms. In the case of the simulator studied here, we also learn about epidemiological processes, the interaction between epidemiologically meaningful parameters and their plausible values, which can then be compared to those found in the literature. Finally, history matching can provide large numbers of calibrated input values, which can be used to run the simulator into the future, allowing predictions to incorporate the uncertainty about the simulator's input values. In our case, calibrated input values are fed into several other research projects, one of which is [14], a complex decision analysis on predicting the costs and effects of different ways of scaling-up access to HIV treatment in Uganda.

This paper is structured as follows: Section 2 describes the individual-based simulator. Section 3 gives an overview of history matching and details the methodological contributions of this paper: the use of linear regression models as emulators and the sampling algorithm. Section 4 shows the fit of the outputs to the observations, notes the reduction of the non-implausible space, and presents key conclusions on the simulator's behavior that were drawn from history matching. It also discusses the benefits of linear regression–based emulators and the effectiveness of the proposed sampling scheme. Section 5 concludes this work.

2. Simulator and problem description. The simulator we developed was an individual-based model, written in NetLogo [31]. It simulates births, deaths, and population growth between 1950 and 2030, the formation and dissolution of sexual partnerships, HIV transmission, disease progression and mortality, the HIV/ART care pathway, the effects of ART on HIV mortality and transmission, and the development and transmission of drug resistance. Key pathways into and through care are explicitly simulated, to allow the effects of different ways of scaling up ART coverage to be accurately captured. In the simulator, people can be tested for HIV through routine, intervention, or antenatal HIV testing programs, or after experiencing HIV-related morbidity. Upon testing positive, a proportion of people are successfully linked to care. Once linked to care, a proportion of people who are eligible start ART, and the remainder receive pre-ART care. People can move from pre-ART care to ART once they are identified as eligible following a CD4 test, or due to severe morbidity. People can drop out of care at any stage. People who drop out of pre-ART care can re-enter through the same pathways used by people to enter initially. People who drop out of ART then restart ART at a rate determined by input parameters. Changes in ART eligibility criteria over time in Uganda are also simulated. In total, 50 simulator outputs and acceptable ranges were selected to enable the model, once calibrated, to accurately reflect key features of HIV epidemiology and care in Uganda. These consist of the following:

- Four demographic outputs, which captured key aspects of non-HIV mortality and population growth in Uganda.
- Nine sexual behavior outputs, which captured patterns of sexual behavior in the coun-

try.
- Five HIV prevalence outputs, to ensure that the model reflects trends in male and female HIV prevalence in Uganda over time.
- The median survival with HIV before the introduction of ART.
- Eight HIV testing outputs, reflecting trends in rates of HIV testing in HIV positive and negative men and women over time.
- Four pre-ART care coverage and twelve ART coverage outputs, to capture the scale-up of HIV care in Uganda in 2003–2014.
- Five ART retention outputs, to capture ART drop-out and restart rates.
- Three second line ART outputs, to capture the proportion of people on second line ART.

The simulator was designed and parameterized to represent the population of Uganda as a whole. To improve simulator run times, however, only 1/2000th of the population of Uganda was simulated in each model run. As a result, we were interested in modeling only the mean output of the simulator, while the variance was deemed to have no real-world meaning in this case.

Once calibrated, the model can be used to simulate different ART scale-up strategies, and to estimate their effects on HIV incidence, mortality, morbidity, and drug resistance and on the ART program and other healthcare costs. A full list of the simulator's inputs and outputs is given in the supplementary material (MukSupplement.pdf [local/web 222KB]). A detailed description of the simulator can be found in [14].

3. Methods.

3.1. History matching.

History matching assumes the existence of a physical process y, which is observed through observations z, and a computer model (simulator) that models y. The goal of history matching is to identify the regions of input space corresponding to acceptable matches, and this is performed by ruling out the *implausible* regions iteratively in waves. A brief description of history matching is given in the following, and a more detailed exposition can be found in [26, 2].

3.1.1. Linking the emulator to observations.

Let

$$z = y + \phi, \tag{1}$$

where z is an observation of the physical process y which is done with some measurement error ϕ. Also let

$$y = g(\mathbf{x}^*) + \delta, \tag{2}$$

where $g(\mathbf{x}^*)$ is the simulator's output when this is evaluated at input $\mathbf{x}^*$. The term δ is the model error term, which represents the discrepancy between the simulator's output when this is evaluated at its "best" input $\mathbf{x}^*$ and the physical process y. This discrepancy often arises because either some parts of the physical process are not completely understood, and it is not possible to include them in the simulator, or they have been deliberately left out, e.g., for mathematical or computational tractability. For more on model discrepancy the reader may consult [11, 8, 5].

Evaluating $g(\mathbf{x})$ can be time-consuming, and exploration of the simulator's input space can require a very large number of evaluations. For this reason, a surrogate statistical model (*emulator*) is built for the simulator, which (a) can predict $g(\mathbf{x})$ for any $\mathbf{x}$ of interest very quickly and (b) can quantify the uncertainty of its predictions. We will return to emulation in section 3.2, but for the moment let us just say that the emulator's predictions are linked to $g(\mathbf{x})$ via

$$g(\mathbf{x}) = \mathrm{E}^*[g(\mathbf{x})] + \zeta(\mathbf{x}), \tag{3}$$

where $\mathrm{E}^*[g(\mathbf{x})]$ is the emulator's prediction for $g(\mathbf{x})$ and where $\zeta(\mathbf{x})$ is the estimation error, whose statistical characteristics can vary with $\mathbf{x}$.

Combining (1), (2), and (3), we can write

$$z = \mathrm{E}^*[g(\mathbf{x}^*)] + \zeta(\mathbf{x}^*) + \delta + \phi. \tag{4}$$

The above equation refers to a single-output simulator and accordingly to a scalar observation. In case the simulator has R outputs and there are R observations z available, there will be R instances of (4), where each quantity will be indexed by the output index r.

3.1.2. The implausibility measure. History matching works by rejecting any input space which is found to be implausible. This characterization is done using the *implausibility measure,* which is based on (4). For the rth output we can write the variance of the error terms in (4) as $V_{o,r} = \mathrm{Var}[\phi_r]$, $V_{m,r} = \mathrm{Var}[\delta_r]$, and $V_{c,r} = \mathrm{Var}[\zeta_r(\mathbf{x})]$. We can then formulate a natural metric for the distance between the observation z_r and the emulator's prediction at $\mathbf{x}$ as

$$I_r(\mathbf{x}) = \frac{|z_r - \mathrm{E}^*[g_r(\mathbf{x})]|}{(V_{o,r} + V_{m,r} + V_{c,r})^{1/2}}. \tag{5}$$

This is the basic form of the implausibility measure for one output, which is essentially the distance between z_r and $\mathrm{E}^*[g_r(\mathbf{x})]$, standardized by all the uncertainties that might be present in the system: the uncertainty due to observation error $V_{o,r}$, model error $V_{m,r}$, and the code uncertainty $V_{c,r}$, which arises because we cannot evaluate the simulator (code) for every $\mathbf{x}$ and we substitute it with the emulator.

Simple distributional assumptions on the form of the various error terms, namely a zero mean and a unimodal distribution, allow us to use the powerful and underused Pukelsheim's rule [21] to derive cut-off values for the implausibility, that is, to come up with thresholds such that if $I_r(\mathbf{x})$ exceeds them, we can be fairly confident that the simulator's output $g(\mathbf{x})$ will not be close to the observations z for this particular value of $\mathbf{x}$. Pukelsheim's 3-sigma rule states that any unimodal continuous distribution contains 95% of its probability mass within three standard deviations from its mean, regardless of its skewness or higher moments. Therefore, for $I_r(\mathbf{x}) > 3$ it will be highly unlikely that the simulator's rth output will match the respective observation for that particular $\mathbf{x}$.

A simple extension of the single output implausibility (5) to multiple outputs can be found by maximizing across all outputs, i.e.,

$$I(\mathbf{x}) = \max_r I_r(\mathbf{x}).$$

The implausibility measure has several other extensions, some of which can incorporate correlation structures between outputs. For more information, the interested reader is referred to the detailed discussion in [26].

3.1.3. Procedure. History matching iteratively discards parts of the input space which are calculated as implausible and therefore highly unlikely to contain matches between the simulator's outputs and the observations. In wave η, the search for acceptable matches is limited to the previous wave's nonimplausible space ($\mathcal{X}_{\eta-1}$), and as a result the nonimplausible space shrinks with each iteration (i.e., $\mathcal{X}_\eta \subset \mathcal{X}_{\eta-1}$). An outline of the procedure is given in the following:

1. Define the initial P-dimensional nonimplausible space $\mathcal{X}_{\eta=0}$.
2. Select N training points from the current nonimplausible space $\mathcal{X}_\eta$, using a space-filling design or some other method that aims to cover $\mathcal{X}_\eta$.
3. Evaluate the simulator at each of the N points. If the model is stochastic, run the simulator K times at each of the N points. Denoting by $\hat{g}(\mathbf{x})$ the averaged simulator output evaluated at $\mathbf{x}$, form the training data $D = \{\mathbf{x}_n, \hat{g}_n(\mathbf{x})\}_{n=1}^N$.
4. Build and validate an emulator for as many of the simulator's R outputs as is possible. Denote the set of emulated outputs as $R_{\eta+1}$. The emulators of wave $\eta+1$ are defined only over $\mathcal{X}_\eta$ and should be more accurate than the emulators of the previous wave, as $\mathcal{X}_\eta$ is smaller than $\mathcal{X}_{\eta-1}$.
5. Evaluate the implausibility measure $I(\mathbf{x})$ over all $r \in R_{\eta+1}$ for a large number of $\mathbf{x} \in \mathcal{X}_\eta$ such that $\mathcal{X}_\eta$ is represented with sufficient accuracy. $\mathcal{X}_{\eta+1}$ is defined as the set of $\mathbf{x} \in \mathcal{X}_\eta$ for which $I(\mathbf{x})$ is less than a chosen threshold. $\mathcal{X}_{\eta+1}$ should be smaller than $\mathcal{X}_\eta$.
6. Unless one of the following conditions is true, increase wave counter η by 1 and repeat steps 2–5:
   (a) The emulator's uncertainty V_c is smaller than the other uncertainties (e.g., V_o or V_m); therefore more waves would most likely lead to little further reduction of $\mathcal{X}_\eta$.
   (b) All $\mathcal{X}_\eta$ is implausible (i.e., all $\mathcal{X}_{\eta+1}$ is empty).
   (c) A sufficient number of points $\mathbf{x}$ that match the observation data have been collected for the purposes of subsequent analyses.

Some comments on the above procedure: in step 2, a reasonable method for choosing the N points at which the simulator is to be evaluated is a uniform design with some space-filling properties. Were we to know the particular type of regression that we would fit, perhaps alternative designs could be more appropriate. But in the absence of such information, a uniform, space-filling design is a good all-purpose choice that is informative about the whole space. A standard choice for creating such a design is via a maximin Latin hypercube [16], and that method can be used when possible. This design, however, is challenging to create in high dimensional spaces of arbitrary shapes, as $\mathcal{X}_\eta$ is likely to be. A simple but effective alternative is the following: start with a large number of points distributed uniformly in $\mathcal{X}_\eta$, e.g., as provided by the slice sampler of section 3.3. Choose the first point at random, and choose the second as the one that is the furthest apart from the first, in the sense of the Euclidean distance. For each of the remaining points, calculate the distance to the closest

of the two first points (minimum distance) and choose as the third the one with the largest minimum distance to the first two (i.e., maximum minimum distance—maximin). Similarly, choose as fourth the point with a maximin distance to the first three, and so on until N points are collected. This is a simple procedure that returns points that are sufficiently well-spread and cover the entire input space, assuming that enough samples from $\mathcal{X}_\eta$ are available.

Condition 6a implies that decreases in the V_c term (code uncertainty), which should come with additional waves and improved emulators, are unlikely to contribute to shrinking $\mathcal{X}_\eta$ further, as the denominator of the implausibility $I_r(\mathbf{x})$ will be dominated by V_o and V_m. If condition 6a occurs, most of the simulator runs should fall within the observations, and history matching can be stopped. At this point, sampling the nonimplausible space $\mathcal{X}_\eta$ should provide as many input parameters that match the observations as required by the application.

Condition 6b is an indication that the simulator cannot match the observations, unless the errors V_o, V_m are revised and perhaps increased. Flagging a simulator's possible inability to match a particular calibration data set is a strong point of history matching, which is contrasted to more traditional Bayesian calibration approaches that will return an input parameter posterior distribution regardless of the quality of the match.

Regarding step 3, outputs that are hard to emulate in the initial waves, perhaps due to the large variation of the inputs, can become easier to handle in later waves, when the inputs are confined in more interesting input space parts, where the simulator's response can be smoother. Additionally, inputs that have a strong effect on outputs in the initial waves can become less important in later waves when their range has been reduced, and other, previously unnoticed, inputs can start having a greater impact on the simulator's behavior, allowing more detailed emulator construction.

Finally, the nonimplausible space can be reduced by several orders of magnitude at each iteration, as will be demonstrated in the results section (section 4). As a result, within a very small number of waves, even a very dense initial design can end up having all its points outside the region of interest. Therefore, the strategy of evaluating the simulator at each wave for a relatively smaller number of times allows us to focus the computational effort on input space areas that are more likely to produce a match to the data.

3.1.4. Convergence. History matching continues until one of the three conditions mentioned in the above procedure is satisfied. That is, the history matching waves proceed until the emulator uncertainty is smaller than the other uncertainties in the implausibility measure, or until all $\mathcal{X}_\eta$ have been characterized as implausible, or until enough matches to the observation data have been found for the purposes of the application.

A natural question that can arise at this point is whether history matching will successfully identify all input space regions that match the observation data or whether some areas might be missed. Technically, it is possible to miss some areas of the input space that result in a match, i.e., incorrectly identify them as implausible, if the emulators do not accurately represent the uncertainty about the simulator's behavior. Additionally, the implausibility can be seen as a statistical test that predicts whether a particular input $\mathbf{x}$ will match the outputs to the calibration data. Even conservative (i.e., large) cut-off values imply that there is a small but nevertheless nonzero probability that a good input is left out.

We can guard against the first condition by ensuring that the emulators are validated.

That is, we confirm that the simulator's outputs fall within the uncertainty intervals provided by the emulator. The second condition can be guarded against by choosing suitably high implausibility cut-offs, especially in the initial waves, where the uncertainty about the simulator's behavior is large. Finally, the smooth behavior that a biological system simulator should possess offers additional confidence that input regions of interest have not been left out.

3.2. Linear regression emulators. As mentioned in the introduction, history matching typically relies on emulators to ease the computational burden of having to evaluate a potentially slow simulator a large number of times. Gaussian processes (GPs) have been used extensively to build emulators, as they are a flexible statistical model with broad presence in recent literature. In this work, however, we are using linear regression as the model of choice for building emulators. Even though linear regression models tend to be less flexible than GPs, they do offer some advantages for history matching. First, they are generally easier to fit than GPs, which is of assistance in the presence of a large number of simulator outputs, each of which requires its own emulator. Second, in complex high dimensional simulators it is common that each output is not influenced by every input of the simulator, but there tends to be a subset of inputs that more greatly affects the behavior of a particular output. These inputs are generally referred to as *active inputs*. Active inputs are not always known a priori, and different sets of inputs might affect the same output at different waves, as the input space shrinks due to history matching. Linear regression models offer a simple and established way of choosing the active inputs for each output at each wave. Although similar results could be achieved using GPs and automatic relevance determination (ARD) [34], using linear regression models can be more straightforward. Finally, linear models are considerably easier to implement.

The fundamental equation for linear regression is

$$g(\mathbf{x}) = \sum_{i=1}^{q} h_i(\mathbf{x})\beta_i + \epsilon, \tag{6}$$

where $h_i(\mathbf{x})$ are functions of the inputs $\mathbf{x}$, β_i are their respective coefficients, and ϵ is residual uncorrelated noise. The functions $h_i(\cdot)$ can take any form (linear, quadratic, interaction term between components of $\mathbf{x}$, or other nonlinear transformation). The term "linear" in the description "linear regression model" therefore refers to the linear relationship between the arbitrary functions $h_i(\mathbf{x})$ and the coefficients β_i. Determining the exact form of the $h_i(\mathbf{x})$ functions is essentially fitting the (statistical) model, and the strategy we follow here is presented in section 3.2.1. Denoting $h(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_q(\mathbf{x})]$ and $\beta = [\beta_1, \ldots, \beta_q]^{\mathrm{T}}$, where $[\cdot]^{\mathrm{T}}$ is the vector transpose, we can write (6) as

$$g(\mathbf{x}) = h(\mathbf{x})\beta + \epsilon. \tag{7}$$

At each wave the simulator is evaluated K times at N points, thus producing the training data $D = \{\mathbf{x}_n, \hat{g}(\mathbf{x}_n)\}_{n=1}^{N}$, which we also denote for brevity as $D = (X, Y)$. If H is an $N \times q$ matrix whose rows are the vectors $h(\mathbf{x}_1), \ldots, h(\mathbf{x}_N)$, the maximum likelihood estimate (m.l.e.) of β is given by the well-known equation

$$\hat{\beta} = (H^{\mathrm{T}}H)^{-1}H^{\mathrm{T}}Y.$$

Similarly, the model's prediction at an untested $\mathbf{x}$ is simply given by

$$\mathrm{E}^*[\mathbf{x}] = h(\mathbf{x})\hat{\beta}.$$

Finally, the m.l.e. of the uncertainty about a prediction at a new input $\mathbf{x}$ is given by

$$\hat{\sigma}^2 = (Y^{\mathrm{T}}Y - Y^{\mathrm{T}}H(H^{\mathrm{T}}H)^{-1}H^{\mathrm{T}}Y)/N.$$

The $\hat{\sigma}^2$ term represents the code uncertainty of section 3.1.1. As a more subtle point here we could mention that $\hat{\sigma}^2$ also includes also the uncertainty that arises from estimating $g(\mathbf{x})$ from the averages $\hat{g}(\mathbf{x})$, which in a GP emulator would have been modeled by the nugget term [1].

3.2.1. Fitting strategy. The main tool we use in determining the exact functionals for the $h(\mathbf{x})$ terms is based on the Bayesian information criterion (BIC) [22], which in this case is given by

$$BIC = N[\ln(2\pi\hat{\sigma}^2) + 1] + (q+1)\ln(N).$$

According to this, when presented with two alternative sets of functions $\{h_i(\mathbf{x})\}_{i=1}^{q_1}$ and $\{h'_i(\mathbf{x})\}_{i=1}^{q_2}$, the one that results in a lower score for the BIC is to be preferred. Using this as our main fitting tool we develop the following strategy.

Zero order: Always include a constant term to account for the overall mean of the data. The *current regression matrix* is set to $h(\mathbf{x}) = 1$.

First order: Form the regression matrix $h'(\mathbf{x}) = [h(\mathbf{x}), x_p]$ for each of the P inputs in the model, and compare the resulting BIC to that of the model that includes only $h(\mathbf{x})$. The input p that results in the largest drop in the BIC is added in the emulator's *active input* set, and the matrix $[h(\mathbf{x}), x_p]$ becomes the current regression matrix; i.e., $h(\mathbf{x}) \leftarrow [h(\mathbf{x}), x_p]$. The procedure is repeated for the rest of the inputs and stops when no additional input decreases the BIC further. The inputs included in the emulator in this way form its *active input* set, which is often much smaller than the full input set of the simulator.

Higher orders: Here by higher order terms we mean functions $h_i(\mathbf{x})$ that include either powers of x_p greater than 1 or products of x_p and their higher order terms. We refer to a term as having nth order when the sum of the powers of the terms involved equals n. A second order term, for example, can contain interactions $x_i x_j$ or squares such as x_i^2. We follow two strategies for the higher order terms.

Exhaustive: If the number of all possible combinations of nth order terms is not prohibitively large, these terms are included one by one in the current regression matrix, and the one that results in the biggest drop in the BIC is added. The procedure is followed for all the remaining terms.

Incremental: Suppose that we have a number of $(n-1)$th order terms and we are trying to investigate whether any nth order terms would help improve the model. We create an nth order term by multiplying one $(n-1)$th order term with a first order term that is already included in the model (i.e., active input). We test whether adding this new term into the model reduces the BIC and include it in the current regression matrix if it does or discard it if it does not. We repeat the same procedure for all the remaining combinations of $n-1$

and first order terms. This procedure allows us to find high order terms which can improve the model without having to test all possible high order combinations of powers of $\mathbf{x}$, which very soon become so numerous that their evaluation is prohibitive and can lead to serious overfitting concerns.

Pruning: Before looking into increasing the order of the terms that are included in the model, we check whether some currently included terms could be removed. We do this by removing one term at a time from the current model. If this removal decreases the BIC, the respective term is removed from the current regression matrix. This reduction can help build a more parsimonious model.

Validation: The testing and addition of a large number of polynomial terms can lead to a model that is overfitted. We guard against this with a leave-one-out validation. Once the final form of the regression matrix has been decided, one training data point is left out, and the regression coefficients are calculated using the remaining ones. The prediction errors are then calculated and divided by $\hat{\sigma}^2$. The result is compared to a normal distribution. If the errors do not deviate significantly from a normal distribution, the emulator is declared valid. If not, we remove the highest order terms that are currently included in the model and repeat this until the emulator validates. If the emulator cannot validate despite this model simplification, this particular output is not considered in the current wave. This is an important strength of history matching, as outputs that are more difficult to emulate can be left until later waves, when they may be much easier to deal with.

3.3. Sampling algorithm. After a number of history matching waves, the nonimplausible space is typically a very tiny portion of the initial nonimplausible space $\mathcal{X}_0$, and it can be several orders of magnitude smaller than the *minimum enclosing hyperrectangle*. We define the latter as the smallest hyperrectangle that encloses all known nonimplausible samples. The minimum enclosing hyperrectangle coincides with $\mathcal{X}_0$ at wave 0 and cannot increase from one wave to the next. The shape of the enclosed nonimplausible space is generally unknown and can only be described by the collection of $\mathbf{x}$'s which satisfy the implausibility conditions for *all* emulators across *all* waves. For simplicity we define an indicator function $\mathcal{I}(\mathbf{x})$, which takes the value 1 if $\mathbf{x}$ is nonimplausible for all emulators across all waves and 0 otherwise.

For history matching to advance, it is necessary to have a large enough number of samples for which $\mathcal{I}(\mathbf{x}) = 1$ such that they cover as much of the nonimplausible region as possible. Furthermore, we would like these samples to be uniformly distributed over the nonimplausible region, as we have no reason to favor some of its parts over others, and for design reasons mentioned in the comments of section 3.1.3. In the following we describe a method for drawing uniform samples from such spaces in an efficient manner, based on an adaptation of the slice sampler to the specific requirements of history matching. We stress that the exact shape of the nonimplausible space cannot be described analytically and can only be known via sampling.

The one dimensional slice sampler [13] works as follows: suppose we have a sample x_i from a distribution $p(x)$ and we want to draw another sample from the same distribution. $p(x_i)$ is evaluated, and a sample u is uniformly drawn from the interval $[0, p(x_i)]$. Left and right sampling limits x_l and x_r are proposed and are incrementally expanded until $p(x_l) < u$ and $p(x_r) < u$ are satisfied. This is the "stepping out" part of the algorithm. In the "shrinking" part of the algorithm, a sample x_{i+1} is uniformly drawn between x_l and x_r. If $p(x_{i+1}) > u$,

then x_{i+1} is accepted as the next sample. If not, x_l is set to x_{i+1} if $x_{i+1} < x_i$ or to $x_r = x_{i+1}$ if $x_{i+1} > x_i$, and the process is repeated until $p(x_{i+1}) > u$.

History matching permits two simplifications to the algorithm sketched above. The region of interest for each sample is known and is defined by the limits of the minimum enclosing hyperrectangle. This makes the stepping out part of the algorithm redundant, as we can always set x_l and x_r for each dimension (i.e., simulator input) to those limits. The second simplification comes from the fact that we want to give uniform weights to any point $\{\mathbf{x} : \mathcal{I}(\mathbf{x}) = 1\}$. We can therefore set $p(x) = \text{const}$. This eliminates the need for calculating the uniform number u, and the condition $p(x_{i+1}) > u$ becomes simply $\mathcal{I}(\mathbf{x}) = 1$. That is, if the proposed sample is nonimplausible, it is accepted; otherwise, it is rejected.

The case described above refers to one input. Higher dimensions can be accommodated by updating each dimension sequentially. The new sample $\mathbf{x}_{i+1}$ is accepted when all dimensions have been updated. The algorithm is outlined in the following.

---

s0 Assume the existence of one $\mathbf{x}$ such that $\mathcal{I}(\mathbf{x}) = 1$ and a minimum enclosing hyperrectangle with upper and lower limits for each dimension p, denoted as $x_{p,\max}$ and $x_{p,\min}$, respectively. x_p is the pth element of $\mathbf{x}$.
s1 let $\mathbf{x}' = \mathbf{x}$.
s2 for $p = 1 : P$
s3 set $x_l = x_{p,\min}$, $x_r = x_{p,\max}$
s4 do
s5 set $x'_p \sim \text{Unif}[x_l, x_r]$
s6 if $\mathcal{I}(\mathbf{x}') = 0$
s7 if $x'_p < x_p$, $x_l = x'_p$; else $x_r = x'_p$
s8 while $\mathcal{I}(\mathbf{x}') = 0$
s9 $\mathbf{x}'$ is the new nonimplausible sample. Store, set $\mathbf{x} = \mathbf{x}'$, and go to **s2** for drawing another sample.

Evaluation of the membership function $\mathcal{I}(\mathbf{x}')$ typically requires calculating the implausibility $I(\mathbf{x}')$ using all the emulators of the current and all previous waves: sample $\mathbf{x}'$ is nonimplausible at wave η if it is nonimplausible for that and all the waves that precede it. Although it may seem paradoxical that an emulator declaring $\mathbf{x}$ as nonimplausible can be "overruled" by another one that says it is not, two examples will demonstrate that this can actually happen: (a) A wave 1 emulator will typically have larger code uncertainty compared to a wave η emulator, as the first is trained over the entire input space $\mathcal{X}_0$ and the latter over the smaller $\mathcal{X}_{\eta-1}$. As a result, a point within $\mathcal{X}_{\eta-1}$ is more likely to be rejected by the wave η emulator which is more certain about its predictions. (b) At the same time, a point that is outside $\mathcal{X}_{\eta-1}$ was rejected by definition by the wave 1 or a subsequent wave emulator (it would otherwise belong in $\mathcal{X}_{\eta-1}$). This point might still be considered as nonimplausible by a wave η emulator, as this emulator was trained over only the $\mathcal{X}_{\eta-1}$ region, and its estimates outside this are either very uncertain or not reliable. Therefore, determining whether $\mathbf{x}'$ is nonimplausible normally requires the evaluation of all the emulators that have been built so far.

In our case, this represents a worst-case scenario, and fewer evaluations are required in

practice. In the previous section, we mentioned that an emulator does not include all inputs, but only those considered active for that particular output at that particular wave. When the slice sampler proposes a move across the pth dimension, evaluation of $\mathcal{I}(\mathbf{x}')$ requires invoking only the emulators that include p in their active input list. The remaining emulators need not be used, as their results remain unchanged. Therefore, large computational savings can be achieved if, for every input, a list of all emulators that include input p in their active input list is made, and only these are evaluated when the slice sampler proposes a move across the pth dimension.

Additional savings in computation time can arise if the emulators are ranked according to the space they reject. That is, given a set of nonimplausible samples from the previous wave, we can rank all the emulators of the current wave according to the proportion of samples they reject from higher to lower. When the membership function $\mathcal{I}(\mathbf{x})$ needs to be evaluated, the emulator with the highest rejection rate is invoked first. This can lead to substantially fewer emulator evaluations, as the ones invoked first have a greater probability of rejecting a sample, and once a sample is deemed implausible by any single emulator, there is no need to evaluate it any further. Finally, invoking the emulators of different waves in reverse order (i.e., last wave first) can also help speed up the evaluation of $\mathcal{I}(\mathbf{x})$, as the later wave emulators should be more precise over the current nonimplausible region.

The proposed method with the computational shortcuts described above is quite efficient, requires no tuning (e.g., in contrast to the proposal kernel of the Metropolis–Hastings method), and can successfully sample very small spaces. It is, nevertheless, an MCMC method, and it can still be affected by poor mixing, especially when inputs are highly correlated. The quality of the mixing therefore needs to be evaluated after the sampling is completed, and the chains should be thinned if the mixing is found to be poor.

Additionally, the slice sampler can capture some disconnected regions but not all. As the inputs are updated on a one-by-one basis, the disconnected regions would need to have overlapping projections in all but one input dimension for a jump to be possible. Starting the algorithm from a large number of nonimplausible samples reduces the probability that a disconnected region is not sampled.

In a typical application of history matching, we have a few thousand nonimplausible samples at each wave. In this case, a large number of parallel chains can be run, each one initialized from a nonimplausible sample. The availability of a multicore machine or a multinode high performance computing cluster can significantly speed up the sampling process. The easy parallelization of this method further increases its efficiency and improves the handling of input space features such as disconnected regions.

4. Results. The epidemiological simulator we study has 96 inputs and 50 outputs. A full list of the inputs, along with the initial nonimplausible ranges, is given in the supplementary material (MukSupplement.pdf [local/web 222KB]), as well as a full list of the simulator outputs with the observation data and their ranges. The range for the latter is assumed to represent $\pm 2(V_m + V_o)^{1/2}$, that is, four standard deviations calculated from the sum of the observation and the model error. 3000 training points were chosen for the first four waves to help explore the simulator's behavior, but these were reduced to 1000 from wave 5 onwards, as the resulting emulators were still successful in rejecting input space. The simulator was run for $K = 30$

times at each design point to allow the estimation of its mean output value. The goal of history matching was to find input parameter values that would lead to mean outputs (as opposed to individual runs) that fell within the observation ranges. A total of 13 waves were carried out.

4.1. Output matching. Figure 1 shows ten simulator outputs which come from two different time series: the proportion of HIV positive people on ART and the proportion of people starting ART with low CD4 counts (< 250 cells/μl). The observation ranges are shown using black bars, and the simulator's output at four different waves (1, 4, 8, and 14) is shown using darkening shades of blue. The figure demonstrates that as the input space shrinks, the simulator's output converges to the observations.

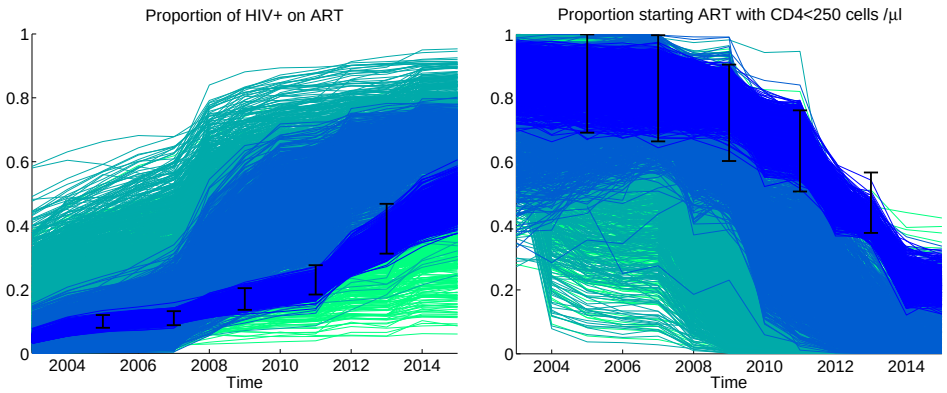

Figure 1. *Ten simulator outputs from two different time series at waves* 1, 4, 8, *and* 14. *The observation ranges for the ten outputs are shown using the black bars. The simulator's output at the four different waves is shown using darkening shades of blue.*

Figure 2 shows histograms of simulator outputs for five different output pairs across waves 1, 8, and 14. The observation ranges are shown with black rectangles. The figure again shows the convergence of the simulator outputs towards the observations as waves progress. It is interesting to note that in the first wave (first column of Figure 2), outputs 17, 18, 45, and 51 are completely off target. In the final wave, the majority of the simulator's outputs are within the targets, with output 26 only being slightly off. Incidentally, this was the output with the poorest matching among all fifty. Histograms of all simulator outputs at wave 14 with their observation ranges are given in the supplementary material (MukSupplement.pdf [local/web 222KB]).

Apart from convergence to the observations, Figure 2 also reveals correlations between the outputs, which are interesting especially in wave 14 (rightmost column). The top right panel shows that there is a strong positive correlation ($r = 0.84$) between the proportion of HIV negative women and the proportion of HIV positive women who have ever been tested in 2011. This reflects two factors. The first is the way that HIV testing rates were controlled in the simulator. One input parameter controlled the absolute rate of testing in HIV negative women. Another controlled the rate of testing in HIV positive women *relative* to the rate in HIV negative women. This introduced a correlation between the two outputs. The second factor is that HIV negative women could become HIV positive through transmission of the

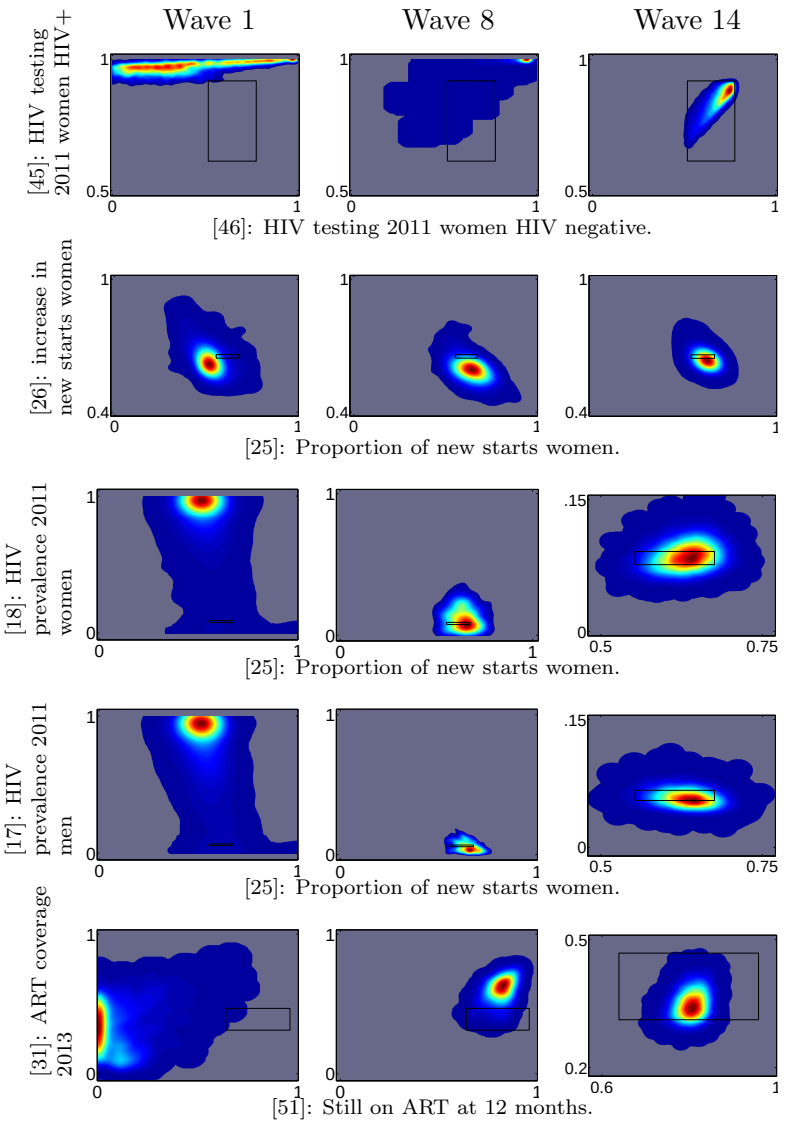

Figure 2. *Histograms of simulator outputs for five different output pairs across waves* 1, 8, *and* 14. *The calibration targets are shown with black rectangles. Note the different scale in some panels of the rightmost column. Numbers in square brackets refer to the simulator's input number.*

virus. Women who were tested for HIV when they were still uninfected would remain "ever tested" after becoming HIV positive. This introduced a further correlation between the two outputs.

In Uganda, the numbers of women starting ART each year are higher than the numbers of men starting. This is due both to the higher prevalence of HIV in women and to the fact that they are more likely to be diagnosed (e.g., through antenatal HIV testing). From

2013, a change in policy meant that all HIV positive pregnant women became eligible for treatment, regardless of how far their disease had progressed. This increased the numbers of women starting ART. We therefore calibrate the simulator to the proportion of people newly starting ART in 2010 who were women (output 25) and the increase in the proportion of new starts who were women between 2010 and 2014 (output 26). There was a clear negative correlation between the two outputs, as shown in Figure 2 ($r = 0.55$). This is because if both outputs had high values, a very high proportion of people starting ART in 2014 would be women. To achieve this, very few men would be able to start ART in 2013, and the large increase in ART coverage between 2011 and 2013 could not be achieved. The proportion of people newly starting ART in 2010 who were women (output 25) was also weakly positively correlated with the HIV prevalence in women in 2011 (output 18) (see Figure 2, $r = 0.21$) and weakly negatively correlated with the HIV prevalence in men in 2011 (output 17; see Figure 2, $r = -0.21$). This reflects the fact that, all else being equal, the proportion of people starting ART who are women will be higher when HIV prevalence in women is high relative to the prevalence in men.

Finally, there is a positive correlation between ART coverage in 2013 (output 31) and the proportion of people who remain on ART 12 months after first starting it (output 51) (see Figure 2, $r = 0.22$). This is because ART coverage will fall as people stop taking it. The correlation is only weak, however, as the number of people newly starting ART has a larger effect on overall coverage than the rate at which people drop out. It can therefore be seen that the results of history matching can give useful insight into the model's structure.

Figures 1, 2, and the histograms in the online material offer evidence that the final simulator runs match the observations. We can quantify this evidence using the *simulator run implausibility* [26, 2], which quantifies how close an *actual* simulator run is to the observations. For the rth output we define this measure as

$$I_{\mathcal{R},r}(\mathbf{x}) = \frac{|z_r - \hat{g}_r(\mathbf{x})|}{(V_{o,r} + V_{m,r} + \hat{s}^2(\mathbf{x})/K)^{1/2}}, \tag{8}$$

where $\hat{g}(\mathbf{x})$ is an estimate of the simulator's mean and $\widehat{s^2}(\mathbf{x})$ an estimate of its variance evaluated at $\mathbf{x}$ and calculated using actual simulator runs. The rest of the terms were defined in section 3.1. Equation (8) is similar to the implausibility measure defined in section 3.1.2. The difference is that this term does not involve any emulators and is a metric that quantifies how close the mean of the rth simulator's output is to the observations for a particular $\mathbf{x}$. Also, (8) is not part of the history matching algorithm but is just a convenient way of evaluating the closeness of the simulator's outputs to the observation data.

The simulator was evaluated 30 times at 22000 different nonimplausible inputs at wave 14. The measure in (8) was evaluated for each of the 50 simulator outputs and each of the 22000 runs. Figure 3 shows the percentage of those runs that had $I_{\mathcal{R},r}(\mathbf{x}) < 2$. This can be interpreted as runs that would fall within the observation ranges roughly 95% of the time if the distribution of the individual simulator runs (repetitions) for a fixed $\mathbf{x}$ followed a normal distribution. The results show that, for half of the outputs, more than 95% of the 22000 runs had $I_{\mathcal{R},r}(\mathbf{x}) < 2$. This percentage was higher than 80% for 44 out of the 50 outputs. For six outputs the scores were as follows: 14: 49%, 15: 43%, 16: 43%, 17: 54%, 18: 59%,

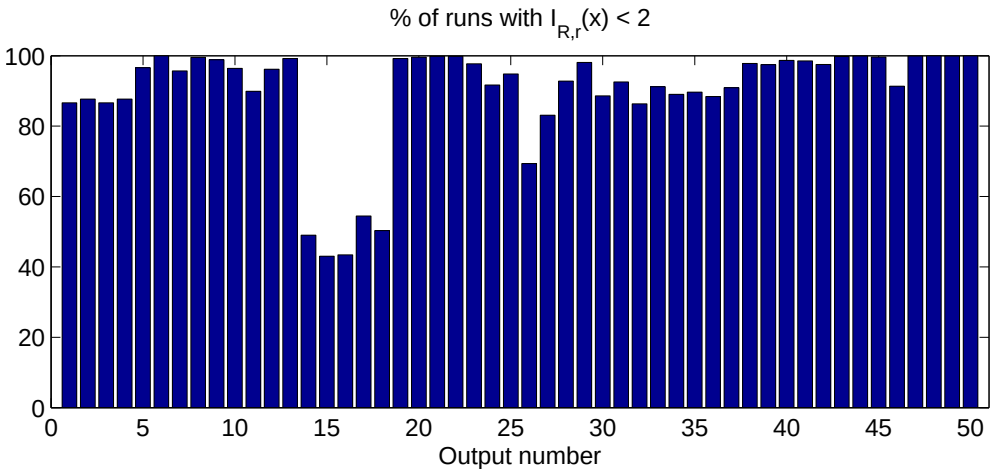

Figure 3. *Percentage of* 20000 *wave* 14 *simulator runs with a simulator run implausibility that is less than* 2*, which can roughly be interpreted as the estimated mean of the simulator's output falling within the observation interval.*

and 26: 69%. This means that although history matching indicated that all these outputs would fall within or just outside the observation ranges, this was true only between 40 and 60% of the time for five outputs and around 70% for the sixth. These outputs were hard to emulate using linear models, in the sense that the prediction uncertainty V_c would not drop beyond a certain magnitude, which was comparable to that of the other two error terms V_m and V_o. The difficulty in emulating these particular outputs is not surprising, as the majority of these outputs represented male and female HIV prevalences at different time points, which are outputs that depend on a large number of inputs and their interactions. Output 26 was highly stochastic (i.e., the samples of individual simulator runs had a large variance), which implies that more simulator evaluations per design point would be needed to increase the accuracy in the estimation of the means and the subsequent emulation.

Not being able to emulate an output is not fatal for history matching. It simply means that fewer simulator runs will be close to the observations than the emulators predicted. If the emulators have been set up and validated correctly, an inaccurate emulator should still not miss the good runs if its uncertainty properly covers the training and validation data. In other words, an emulator that is very uncertain will be unable to rule out regions of input space that actually contain bad matches, but should not incorrectly rule out regions containing matches that are acceptable.

4.2. Input space shrinkage. In this section we examine the shrinking of the input space during the course of waves and present the main epidemiological conclusions that were extracted. The minimum enclosing hyperrectangle at wave 13 was 10^{-33} times smaller than the initial nonimplausible space $\mathcal{X}_0$. This is a very small number, which, however, arises from the large number of inputs and the multiple constraints imposed by the simulator's outputs. Even within this hyperrectangle, however, a tiny proportion of points was nonimplausible. The calculated volume of the nonimplausible space was $\approx 10^{-45}$ times smaller than $\mathcal{X}_0$. That

Table 1

Ratio of the nonimplausible space volume at each wave to the initial nonimplausible space $\mathcal{X}_0$. This table also expresses the probability of finding a nonimplausible sample at wave n if we randomly draw samples from $\mathcal{X}_0$.

Wave 1	$1.8 \cdot 10^{-05}$	Wave 8	$5.2 \cdot 10^{-30}$
Wave 2	$2.6 \cdot 10^{-08}$	Wave 9	$4.5 \cdot 10^{-33}$
Wave 3	$1.6 \cdot 10^{-09}$	Wave 10	$1.2 \cdot 10^{-35}$
Wave 4	$1.7 \cdot 10^{-10}$	Wave 11	$2.9 \cdot 10^{-37}$
Wave 5	$5.2 \cdot 10^{-14}$	Wave 12	$2.9 \cdot 10^{-41}$
Wave 6	$7.7 \cdot 10^{-20}$	Wave 13	$2.4 \cdot 10^{-45}$
Wave 7	$1.1 \cdot 10^{-24}$		

is, only 1 point in 10^{12} (one in a trillion) is nonimplausible if selected at random between the narrowest limits suggested by the last wave's nonimplausible samples. The ratio of the volumes of the final nonimplausible space and $\mathcal{X}_0$ are shown in Table 1. Note that the nonimplausible region at wave 13 is substantially smaller than those found in previous history matching applications in the literature.

The range of 5 out of 96 inputs was reduced to less than 1% of the original, while for around 25 it decreased to less than 50%. For approximately 50 inputs, the range remained similar to the original. Either these inputs do not substantially affect the history matched outputs of the simulator, or there are combinations of these inputs with others that are implausible but cannot be visualized in the one dimensional projection of the nonimplausible space that these histograms represent. The supplementary material (MukSupplement.pdf [local/web 222KB]) includes histograms of the nonimplausible samples at wave 13 for all 96 inputs, which demonstrate the overall input space reduction.

We now focus our attention on a small set of inputs, track their shrinkage through the waves, and draw some conclusions based on their correlation patterns. The lower triangle of the lattice in Figure 4 shows scatter plots of nonimplausible samples for pairs of inputs across four waves. The light blue color is the initial nonimplausible region, and waves 1, 4, 8, and 13 are shown in darkening shades of blue. The baseline transmission [55] (input 55) range is reduced to a third as early as wave 2, and by wave 5 it is down to 10% of the original range. Similar conclusions can be drawn about the other inputs. The upper triangle of the lattice shows two dimensional histograms of wave 13 nonimplausible points as a function of pairs of inputs. The color scale represents the log 10 probability of finding a nonimplausible sample if we fix the values of the inputs that lie across the axes to a particular value and choose the rest of the inputs randomly. For example, the red region in the panel that corresponds to inputs 75 and 55 means that fixing these inputs to those values and varying the others freely gives us a 10^{-43} probability of finding a nonimplausible sample. The grey area of these plots indicates that, for those values of the respective inputs, no samples that match the simulator's output to the observations were found. All the axes in Figure 4 correspond to the initial range of the inputs, as is shown in the supplementary material (MukSupplement.pdf [local/web 222KB]).

Figure 5 is a zoomed-in version of some of the histograms shown in Figure 4 to allow for a more detailed analysis of the correlation patterns. The left panel of Figure 5 shows a histogram of nonimplausible samples for the baseline HIV transmission probability (input 55)

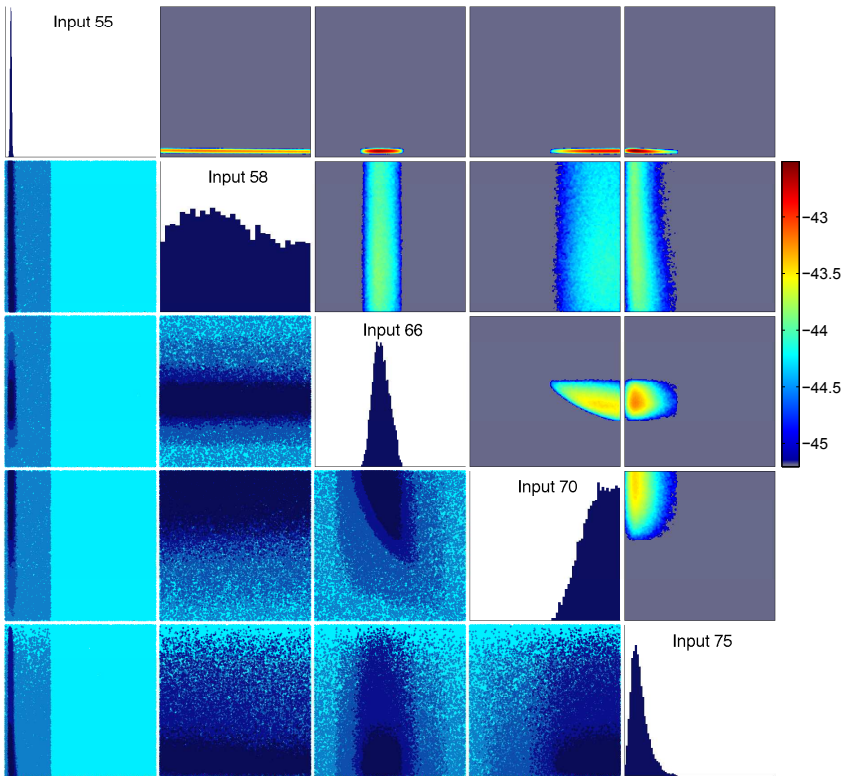

Figure 4. *Summary of the input space shrinking across waves: The lower triangle of the above lattice shows pair plots of nonimplausible samples for five different inputs at waves* 1, 4, 8, *and* 13, *in darkening shades of blue. The upper triangle shows an estimate of the* $\log 10$ *probability of finding a nonimplausible sample after fixing the respective input pairs to a particular value. The gray area indicates that it is virtually impossible to obtain a match for these values of the input pairs. The diagonal shows one dimensional histograms of the wave* 13 *nonimplausible samples for the respective inputs. All axes range between the initial minimum and maximum value of each input.*

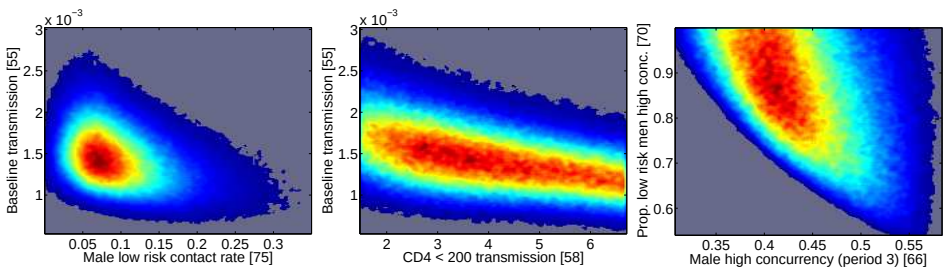

Figure 5. *Histograms of nonimplausible samples at wave* 13 *showing correlation patterns between inputs. See text for an analysis.*

and the rate at which men in one of the two sexual behavior risk groups form new partnerships during a particular time period (input 75). The overall range of input 55 is constrained to between 0.0005 and 0.0027, despite the broad initial plausible range of 0–1. This final range

is consistent with empirical data from Uganda, which estimated the per-sex-act transmission probability to be 0.0011 (95% CI 0.00080–0.0015) [30]. The plausible range for the contact rate (input 75) in the final wave was constrained to be between 0 and 0.3, a large reduction from its original plausible range of 0–1. There is a clear negative correlation between the two inputs ($r = -0.36$), demonstrating that fits are unlikely to be found when the rate at which new partnerships form (and therefore the amount of sex occurring in the model) and the per-sex-act transmission probability are both high or both low.

The middle panel of Figure 5 shows the final wave distribution of the baseline transmission probability (input 55) against the relative increase in transmission probability for people with low CD4 counts (advanced infection) (input 58). Unlike the baseline transmission probability, the overall range of the latter input parameter did not change during calibration, indicating that model fits can be found throughout the initial plausible range. The figure shows that there is a negative correlation between the two input parameters. This occurred as, all else being equal, increasing the value of one parameter and simultaneously decreasing the value of the other will result in similar overall levels of HIV transmission in the model.

Finally, the right panel of the same figure shows the final wave distribution of the proportion of (low risk) men who were able to be in more than one partnership at the same time (input 70) against the associated concurrency input parameter (input 66). The purpose of the concurrency parameter was to influence how likely it was that these men who *could* form additional partnerships *would actually* do so. The graph shows that model fits were unlikely to be found both when the proportion of men who could form additional partnerships was low and when it was not very likely that men who could form additional partnerships would do so. This is because the model was calibrated to sexual behavior data from Uganda that indicates that around 9% of men aged 15–49 have more than one ongoing partnership at any point in time [15].

4.3. The case for linear models. Gaussian processes (GPs) have been extensively used for building computer model emulators in the context of history matching and beyond. GPs are very flexible statistical models, but at the same time more complex and less universal and less understood than the ubiquitous linear regression, which we employ in this work. We make the case here that linear models can be useful in history matching and that they can go a long way toward calibrating high dimensional simulators. Their simplicity and widespread usage can also have some advantages over GPs. Moreover, there is no binary decision that has to be made (i.e., use linear regression or GPs), as both statistical models can be used in the same history match. For example, linear regression can be used at the initial waves if it is found to efficiently reject large portions of the input space, and GP-based emulators can be introduced at later stages if linear regression fails to provide emulators of sufficient accuracy to reduce space further.

As an example, we show results from two emulators built for output 15 at wave 7 using 1000 training points. The first is a linear regression emulator containing 33 terms up to third order. The second is a GP-based emulator, with a third order polynomial mean function and the Matérn correlation function. The GP correlation function parameters (correlation lengths) were estimated from the data by maximizing their likelihood. The GP's mean function parameters (regression coefficients) were integrated out. For more details on this type of GP

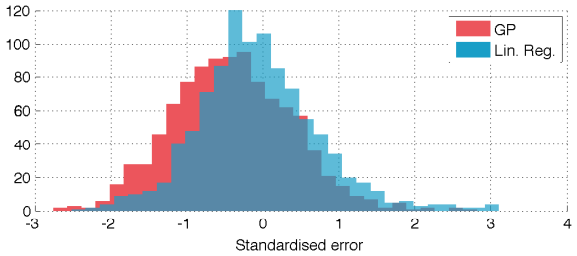

Figure 6. *Standardized errors for the GP and linear regression–based emulators for predictions of the wave 8 simulator runs. Most errors lie within the* $[-2, 2]$ *interval. The GP emulator's errors have a slight negative bias, and the linear regression emulator errors are slightly skewed towards positive values.*

emulator, the reader can consult [2]. Estimating the correlation lengths on the GP emulator took a little more than an hour, while building the linear regression model required a few seconds. The computational load in the estimation of the correlation lengths was due to the optimization algorithm that looked for a mode in a 96-dimensional likelihood function (i.e., one dimension per simulator input). Moreover, it is possible that the optimization algorithm found a suboptimal mode, as the likelihood is almost certainly multimodal. Finding a good mode among several would increase the computational load, as a more detailed exploration of the likelihood surface would be required.

Both the linear regression and the GP-based emulators were then used to predict the simulator's output for the wave 8 runs. Histograms of the standardized errors (i.e., the difference between the simulator's output and the emulator's prediction, divided by the emulator's standard deviation for the prediction [4]) are given in Figure 6. The standardized errors take values mostly in the region $[-2, 2]$, an indication that both emulators are valid. The GP-based emulator, however, resulted in larger code uncertainty (the $V_{c,r}$ term in (5)) when evaluated at the wave 8 nonimplausible samples. As a result, the calculated implausibility was smaller, and it rejected 7% of the wave 8 nonimplausible samples compared to 15% for the linear regression–based emulator. We should also note here that calculating the implausibility for ≈ 20000 wave 8 nonimplausible samples took a few milliseconds for the linear regression emulator and around 15 seconds for the GP-based emulator. This is because the GP-based emulator needs to create an $N \times Np$ correlation matrix, where $N = 1000$ are the training and $Np = 20000$ were the testing points.

As a second example, we used GPs to model the residual between the linear model's predictions and the simulator's outputs in wave 13. That is, (6) is now changed to

$$g(\mathbf{x}) = \sum_{i=1}^{q} h_i(\mathbf{x})\beta_i + \eta(\mathbf{x}),$$

where $\eta(\mathbf{x})$ is a zero mean GP instead of the uncorrelated noise error term ϵ of (6). The rationale here is that some correlation must still exist in the linear regression model's residual ϵ, and capturing this with a GP should reduce the overall uncertainty. Indeed, modeling the residual with a GP resulted in a 22% further shrinkage of the nonimplausible space compared to using the linear regression models alone. This example shows that the two models can

be combined within history matching to increase the rejection rate at the expense of the additional computational cost of training a GP.

GPs are clearly more flexible models and will outperform linear regression in low dimensional regression problems. In high dimensions, however, it is very difficult to have a sufficient number of training points such that the GP can accurately describe the simulator's response surface. As a result, the performance gap between the GPs and the less flexible linear regression becomes smaller. This theoretical argument can support to some extent the usefulness of linear regression–based emulators in history matching of large models. Furthermore, from our experience, a major stumbling block in the adoption of history matching by practitioners has been the requirement to understand and implement a GP-based regression model. Demonstrating that history matching can be carried out using a much simpler and better understood model such as linear regression can increase its adoption as a useful tool for analyzing and calibrating complex models.

4.4. The sampling algorithm. In this section we evaluate the performance of the sampling algorithm and compare it with a simple Metropolis–Hastings (MH) sampling scheme. The MH algorithm uses a transition kernel that is a zero mean multivariate normal with a covariance matrix estimated from 1000 nonimplausible samples of the current wave, scaled to result in an acceptance rate of approximately 25%. The target distribution was uniformly defined over all nonimplausible space. Per sample, the slice sampler requires roughly $P = 96$ times more emulator evaluations, because the inputs are updated sequentially. To allow for a fair comparison the Markov chains in the MH algorithm are run longer, such that the emulator evaluations between the two algorithms are roughly equal. The results are evaluated using the effective sample size (ESS) for each chain and input, which was calculated with the `effectiveSize` function from the R package CODA [20]. This function provides an estimate of the number of samples that can be considered uncorrelated from a Markov chain. Both algorithms were compared at waves 4, 7, 11, and 13 using 1000 nonimplausible samples as starting points; i.e., 1000 chains were run for each case. The ESS scores were averaged across the 1000 chains for each input.

Figure 7 shows the averaged ESS for the MH and the slice sampling algorithms for waves 4, 7, 11, and 13. The ESSs are sorted in increasing order to facilitate the comparison. The figure demonstrates that, in general, the slice sampler results in samples that are less correlated for the same amount of computational effort, often by a very large margin. The only exceptions are inputs 55, 58, and 1 in wave 13 and input 55 in wave 11. The lowest ESS score in wave 13 for the slice sampler was 5 and for the MH scheme was 15. For wave 11, the worst components in both cases had an ESS of around 10. The low ESS scores of the slice sampler were most likely due to the fact that these particular inputs were very correlated, and the correlation information, which was provided to the MH algorithm, was not available to the slice sampler. Providing the slice sampler with this information or using an extension such as [18], could help improve mixing. For the inputs that were affected the most, we have tried to mitigate the low ESS scores by drawing large numbers of samples and visually verifying that the sampled inputs spanned the entire range of nonimplausible samples. Overall, the proposed sampling algorithm gave reasonably good results with the additional benefit of requiring no tuning or manual intervention, while being trivial to implement once the implausibility function is coded up.

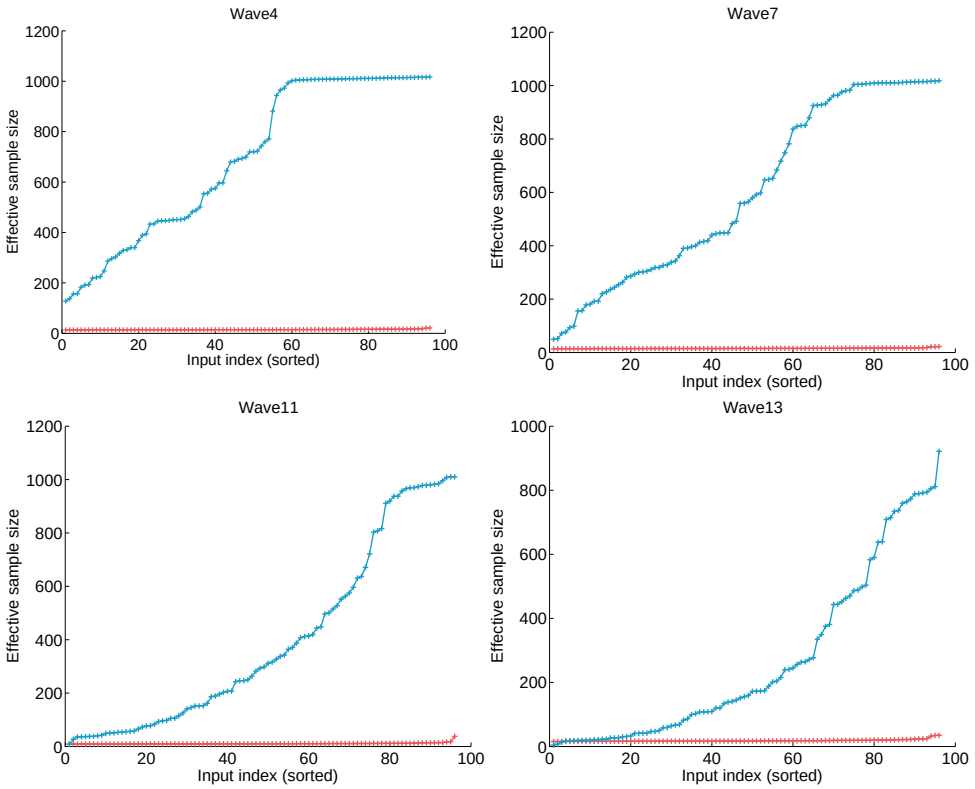

Figure 7. *Averaged ESS for the slice sampler (blue) and the MH (red) algorithms at four different waves. The ESSs were averaged across* 1000 *different chains. The slice sampler chains contained* 1000 *samples each, and those of MH contained the number of samples required to match the computational effort of the slice sampler in terms of emulator evaluations. Each point in the four panels above corresponds to one of the* 96 *inputs, with their indices sorted to facilitate comparison. The slice sampler produced chains with less correlation, as indicated by the higher effective sample size, while in some cases the chains were nearly uncorrelated (effective sample size of* $\sim$ 1000 *in a* 1000 *sample chain). In the case of highly correlated inputs in later waves, the performance of the two algorithms was similar, although the MH algorithm was aware of the correlation structure, while the slice sampler was not.*

5. Conclusion. History matching is a (pre-)calibration method capable of finding parts of a simulator's input space that are likely to match the observations. We have applied this method to a simulator that is larger than any other to which history matching has previously been applied. This scaling up was facilitated by the use of linear regression models as emulators and a sampling algorithm that was capable of sampling high dimensional and very small nonimplausible spaces.

The calibrated simulator was an HIV stochastic individual-based model with 96 inputs and 50 outputs. The simulator's input space was reduced by a factor of 10^{-45} after 13 waves of history matching. In the final wave, the majority of outputs had more than a 90% chance of falling within the observation ranges when the simulator was run at inputs suggested by history matching. Despite the high success for each individual output, getting a simulator run that would match all the observations was relatively rare (around 5 in 1000). However, considering

the size of the problem and the number of outputs, this was still an acceptance proportion that was considered useful for the epidemiologists using the simulator. The simulator could be evaluated approximately 20000 times per day on a high performance computing cluster, and despite the low overall acceptance rate, it was possible to obtain a few hundred runs that simultaneously matched all the observations in reasonable time.

These runs are fed into a number of other research projects, such as [14], that make predictions about the trajectory of HIV in the next 10–15 years, taking into account the uncertainty that is introduced by the simulator's unknown input parameter values. In the past, when making predictions using simulators of this scale, a single input parameter set that would fit the historical data was used, typically found by hand using prior knowledge from the model developer. This approach did not explicitly acknowledge the fact that the simulator's input parameters are uncertain quantities—an uncertainty that was left out of any predictions. The methodology presented here addresses this point. History matching provided us with a few hundred input parameter values, from different parts of the input space, that matched the calibration data, which were then used to run the simulator up to 2030 under 27 different ART scale-up interventions. It therefore offered a method of quantifying the effect of the uncertainty about the input parameter values on the predicted outcome of the ART interventions.

Apart from the large numbers of input values that fit the observations, history matching also provided insights into the simulator's structure. The active input selection methodology revealed the inputs that influenced an output the most, and reductions in the nonimplausible space showed which inputs were affected by the constraints imposed by the observations. Both of these features are very useful in analyzing simulators of this scale. The correlation patterns that emerged between inputs and between outputs highlighted the existence of various structures and processes in the simulator. This information can be used to understand the internal workings of a simulator or, indeed, verify that everything works as intended, knowledge that could lead to the discovery of simulator coding errors, suggest ways in which the simulator can be improved, or even help derive appropriate model discrepancy terms in case the simulator is not capable of matching the observations.

Methodologically, a key feature of this work was the use of linear regression models for building emulators, instead of the GPs that were typically being used in previous history matching applications. Even though linear regression models are less flexible than GPs, they are generally easier to fit, interpret, and implement. Despite their simplicity, they did cover a lot of ground towards calibrating a very complex simulator. This was also facilitated by the history matching philosophy, which does not require an emulator to describe the simulator everywhere in great precision: as long as the simulator runs fall within the uncertainty bounds of the emulator (i.e., the code uncertainty V_c is correctly specified), history matching can proceed. See [27] for further discussions on the topic.

Using a more advanced statistical model for building an emulator can generally reduce the code uncertainty. It is possible, however, especially in the first waves of a history match, that the simulator's outputs $g(\mathbf{x})$ are so far from the observations z that a moderate reduction in the code uncertainty V_c (see (5)) will not have an appreciable effect in reducing the input space further. In later waves, when $\mathrm{E}^*[g(\mathbf{x})]$ and z converge, the effort of building a more sophisticated emulator with smaller V_c could pay dividends. We tried this at the last wave of

our history match, and indeed the GP-based emulator resulted in a further shrinkage of the input space. Hence, we do not try to argue against the use of GPs in building emulators for history matching, but note that linear regression models offer an alternative that is faster and more straightforward to implement.

The availability of a large number of nonimplausible samples is critical in the application of history matching. Sampling the nonimplausible space can be challenging, as this is high dimensional and can be quite small. A simple MCMC algorithm that tackles this problem was proposed in this work, which was simple to implement, requiring virtually no tuning, and was successful in drawing uniform samples from very small nonimplausible spaces. The correlation between some inputs meant that the mixing was slightly poor for a small number of inputs, something that could be addressed using block updating.

In conclusion, the effectiveness and simplicity of the history matching method presented here shows that it is a useful tool for the calibration of computationally expensive, high dimensional, individual-based models.

REFERENCES

[1] I. Andrianakis and P. Challenor, *The effect of the nugget on Gaussian process emulators of computer models*, Comput. Statist. Data Anal., 56 (2012), pp. 4215–4228, https://doi.org/10.1016/j.csda.2012.04.020.

[2] I. Andrianakis, I. Vernon, N. McCreesh, T. J. McKinley, J. E. Oakley, R. Nsubuga, M. Goldstein, and R. G. White, *Bayesian history matching and calibration of complex infectious disease models using emulation: A tutorial and a case study on HIV in Uganda*, PLoS Comput. Biol., 11 (2015), pp. 1–18, https://doi.org/10.1371/journal.pcbi.1003968.

[3] C. Andrieu, A. Doucet, and R. Holenstein, *Particle Markov chain Monte Carlo methods*, J. Roy. Statist. Soc. Ser. B, 72 (2010), pp. 269–342, https://doi.org/10.1111/j.1467-9868.2009.00736.x.

[4] L. S. Bastos and A. O'Hagan, *Diagnostics for Gaussian process emulators*, Technometrics, 51 (2009), pp. 425–438, https://doi.org/10.1198/TECH.2009.08019.

[5] J. Brynjarsdóttir and A. O'Hagan, *Learning about physical parameters: The importance of model discrepancy*, Inverse Problems, 30 (2014), 114007, https://doi.org/10.1088/0266-5611/30/11/114007.

[6] P. S. Craig, M. Goldstein, A. H. Seheult, and J. A. Smith, *Pressure matching for hydrocarbon reservoirs: A case study in the use of Bayes linear strategies for large computer experiments*, in Case Studies in Bayesian Statistics, C. Gastonis, J. S. Hodges, R. E. Kass, R. McCulloch, P. Rossi, and N. D. Singpurwalla, eds., Springer-Verlag, Berlin, 1997, Vol. 3, pp. 37–93, https://doi.org/10.1007/978-1-4612-2290-3.

[7] G. J. Gibson and E. Renshaw, *Estimating parameters in stochastic compartmental models using Markov chain methods*, IMA J. Math. Appl. Med. Biol., 15 (1998), pp. 19–40, https://doi.org/10.1093/imammb/15.1.19.

[8] M. Goldstein and J. Rougier, *Reified Bayesian modelling and inference for physical systems*, J. Statist. Planning Inference, 139 (2009), pp. 1221–1239, https://doi.org/10.1016/j.jspi.2008.07.019.

[9] M. Goldstein, A. Seheult, and I. Vernon, *Assessing model adequacy*, in Environmental Modelling: Finding Simplicity in Complexity, 2nd ed., J. Wainwright and M. Mulligan, eds., Wiley-Blackwell, Chichester, UK, 2013, Chapter 26, https://doi.org/10.1002/9781118351475.ch26.

[10] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient global optimization of expensive black-box functions*, J. Global Optim., 13 (1998), pp. 455–492, https://doi.org/10.1023/A:1008306431147.

[11] M. C. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, J. Roy. Statist. Soc. Ser. B, 63 (2001), pp. 425–464, https://doi.org/10.1111/1467-9868.00294.

[12] J. Knowles, *A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems*, IEEE Trans. Evol. Comput., 10 (2005), pp. 50–66, https://doi.org/10.1109/TEVC.2005.851274.

[13] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, Cambridge, UK, 2003, ISBN: 9780521642989.

[14] N. McCreesh, I. Andrianakis, R. N. Nsubuga, M. Strong, I. Vernon, T. J. McKinley, J. E. Oakley, M. Goldstein, R. Hayes, and R. G. White, *Universal test, treat, and keep: Improving ART retention is key in cost-effective HIV control in Uganda*, BMC Infect. Disease, 17 (2017), 322, https://doi.org/10.1186/s12879-017-2420-y.

[15] N. McCreesh, K. O'Brien, R. N. Nsubuga, L. A. Shafer, R. Bakker, J. Seeley, R. J. Hayes, and R. G. White, *Exploring the potential impact of a reduction in partnership concurrency on HIV incidence in rural Uganda: A modeling study*, Sexually Transmitted Diseases, 39 (2012), pp. 407–413, https://doi.org/10.1136/sextrans-2011-050109.35.

[16] M. D. McKay, R. J. Beckman, and W. J. Conover, *A comparison of three methods for selecting values of input variables in the analysis of output from a computer code*, Technometrics, 21 (1979), pp. 239–245, https://doi.org/10.1080/00401706.1979.10489755.

[17] T. J. McKinley, A. R. Cook, and R. Deardon, *Inference in epidemic models without likelihoods*, Int. J. Biostat., 5 (2009), https://doi.org/10.2202/1557-4679.1171.

[18] I. Murray, R. P. Adams, and D. J. MacKay, *Elliptical slice sampling*, in Proc. 13th Int. Conf. Artificial Intell. Statist., 2010, volume 9, pp. 541–548.

[19] P. D. O'Neill and G. O. Roberts, *Bayesian inference for partially observed stochastic epidemics*, J. Roy. Statist. Soc. Ser. A, 162 (1999), pp. 121–129, https://doi.org/10.1111/1467-985X.00125.

[20] M. Plummer, N. Best, K. Cowles, and K. Vines, *Coda: Convergence diagnosis and output analysis for MCMC*, R News, 6 (2006), pp. 7–11, https://www.r-project.org/doc/Rnews/Rnews_2006-1.pdf.

[21] F. Pukelsheim, *The three sigma rule*, Amer. Statist., 48 (1994), pp. 88–91, https://doi.org/10.2307/2684253.

[22] G. Schwarz, *Estimating the dimension of a model*, Ann. Statist., 6 (1978), pp. 461–464, https://doi.org/10.1214/aos/1176344136.

[23] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. H. Strumpf, *Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems*, J. Roy. Soc. Interface, 6 (2009), pp. 187–202, https://doi.org/10.1098/rsif.2008.0172.

[24] J. United Nations Programme on HIV/AIDS, *The Gap Report*, Geneva, UNAIDS, 2014, http://files.unaids.org/en/media/unaids/contentassets/documents/unaidspublication/2014/UNAIDS_Gap_report_en.pdf.

[25] I. Vernon and M. Goldstein, *A Bayes linear approach to systems biology*, MUCM Technical Report, University of Durham, UK, 2010.

[26] I. Vernon, M. Goldstein, and R. G. Bower, *Galaxy formation: A Bayesian uncertainty analysis*, Bayesian Anal., 5 (2010), pp. 619–670, https://doi.org/10.1214/10-BA524.

[27] I. Vernon, M. Goldstein, and R. G. Bower, *Rejoinder for galaxy formation: A Bayesian uncertainty analysis*, Bayesian Anal., 5 (2010), pp. 697–708, https://doi.org/10.1214/10-BA524REJ.

[28] I. Vernon, M. Goldstein, and R. G. Bower, *Galaxy formation: Bayesian history matching for the observable universe*, Statist. Sci., 29 (2014), pp. 81–90, https://doi.org/10.1214/12-STS412.

[29] I. Vernon, J. Liu, M. Goldstein, J. Rowe, J. Topping, and K. Lindsey, *Bayesian uncertainty analysis for complex systems biology models: Emulation, global parameter searches and evaluation of gene functions.*, BMC Syst. Biol., (2016), submitted.

[30] M. J. Wawer, R. H. Gray, N. K. Sewankambo, D. Serwadda, X. Li, O. Laeyendecker, N. Kiwanuka, G. Kigozi, M. Kiddugavu, T. Lutalo, F. Nalugoda, F. Wabwire-Mangen, M. P. Meehan, and T. C. Quinn, *Rates of HIV-1 transmission per coital act, by stage of HIV-1 infection, in Rakai, Uganda*, J. Infectious Diseases, 191 (2005), pp. 1403–1409, https://doi.org/10.1086/429411.

[31] U. Wilensky, *Netlogo*, 2009, http://ccl.northwestern.edu/netlogo/.

[32] D. Williamson, M. Goldstein, L. Allison, A. Blaker, P. Challenor, L. Jackson, and K. Yamazaki, *History matching for exploring and reducing climate model parameter space using observations and a large perturbed physics ensemble*, Climate Dynam., 41 (2013), pp. 1703–1729, https://doi.org/10.1007/s00382-013-1896-4.

[33] D. Williamson and I. Vernon, *Efficient Uniform Designs for Multi-wave Computer Experiments*, preprint, https://arXiv.org/abs/1309.3520, 2013.

[34] D. P. Wipf and S. S. Nagarajan, *A new view of automatic relevance determination*, in Proc. 20th Conf. on Advances in Neural Inform. Process. Syst., J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, eds., Curran Associates, 2008, pp. 1625–1632.